

WEST Search History

DATE: Thursday, March 17, 2005

Hide?	Set Name	Query	Hit Count
		<i>DB=PGPB,USPT,USOC; PLUR=YES; OP=ADJ</i>	
<input type="checkbox"/>	L12	L11 and global	10
<input type="checkbox"/>	L11	L3 and (off set or offset)	11
<input type="checkbox"/>	L10	L2 and (off set or offset)	92
<input type="checkbox"/>	L9	L3 and (Jar file)	6
<input type="checkbox"/>	L8	L4 and (Jar file)	14
<input type="checkbox"/>	L7	L4 and L1	5
		<i>DB=EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ</i>	
<input type="checkbox"/>	L6	L3	2
		<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ</i>	
<input type="checkbox"/>	L5	L4 and L2	2
<input type="checkbox"/>	L4	portal server	868
<input type="checkbox"/>	L3	L2 and redundant	38
<input type="checkbox"/>	L2	(classfile or class file) and constant pool	217
		<i>DB=PGPB,USPT,USOC; PLUR=YES; OP=ADJ</i>	
<input type="checkbox"/>	L1	717/100-123,151-167.ccls.	4436

END OF SEARCH HISTORY

US-CL-CURRENT: 707/100

ABSTRACT:

A method for packaging program resources includes collecting a set of the program resources that comprises a plurality of object files, which contain data structures having entries that are constants and methods that reference the entries. The data structures in at least some of the object files are combined into a common data pool, in which semantically-identical entries in different ones of the files are represented by a single consolidated entry, irrespective of whether the entries in the different files are syntactically identical. The set of the program resources is packaged together with the common data pool in a combined output file.

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	Know	Draw
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	------

☐ 3. Document ID: US 20030009743 A1

L9: Entry 3 of 6

File: PGPB

Jan 9, 2003

PGPUB-DOCUMENT-NUMBER: 20030009743

PGPUB-FILING-TYPE: new

DOCUMENT-IDENTIFIER: US 20030009743 A1

TITLE: METHOD AND APPARATUS FOR PRE-PROCESSING AND PACKAGING CLASS FILES

PUBLICATION-DATE: January 9, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
FRESKO, NEDIM	SAN FRANCISCO	CA	US	
TUCK, RICHARD	SAN FRANCISCO	CA	US	

US-CL-CURRENT: 717/117

ABSTRACT:

A method and apparatus for pre-processing and packaging class files. Embodiments remove duplicate information elements from a set of class files to reduce the size of individual class files and to prevent redundant resolution of the information elements. Memory allocation requirements are determined in advance for the set of classes as a whole to reduce the complexity of memory allocation when the set of classes are loaded. The class files are stored in a single package for efficient storage, transfer and processing as a unit. In an embodiment, a pre-processor examines each class file in a set of class files to locate duplicate information in the form of redundant constants contained in a constant pool. The duplicate constant is placed in a separate shared table, and all occurrences of the constant are removed from the respective constant pools of the individual class files. During pre-processing, memory allocation requirements are determined for each class file, and used to determine a total allocation requirement for the set of class files. The shared table, the memory allocation requirements and the reduced class files are packaged as a unit in a multi-class file.

☐ 4. Document ID: US 6732108 B2

L9: Entry 4 of 6

File: USPT

May 4, 2004

US-PAT-NO: 6732108

DOCUMENT-IDENTIFIER: US 6732108 B2

TITLE: Class file archives with reduced data volume

DATE-ISSUED: May 4, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Factor; Michael	Haifa			IL
Teperman; Avi	Haifa			IL
Greenwood; Matt	Zichron Yaacov			IL

US-CL-CURRENT: 707/100; 707/1, 707/10, 707/7

ABSTRACT:

A method for packaging program resources includes collecting a set of the program resources that comprises a plurality of object files, which contain data structures having entries that are constants and methods that reference the entries. The data structures in at least some of the object files are combined into a common data pool, in which semantically-identical entries in different ones of the files are represented by a single consolidated entry, irrespective of whether the entries in the different files are syntactically identical. The set of the program resources is packaged together with the common data pool in a combined output file.

72 Claims, 7 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 5

☐ 5. Document ID: US 6530080 B2

L9: Entry 5 of 6

File: USPT

Mar 4, 2003

US-PAT-NO: 6530080

DOCUMENT-IDENTIFIER: US 6530080 B2

TITLE: Method and apparatus for pre-processing and packaging class files

DATE-ISSUED: March 4, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY

Fresko; Nedim San Francisco CA
Tuck; Richard San Francisco CA

US-CL-CURRENT: 717/166; 717/143

ABSTRACT:

A method and apparatus for pre-processing and packaging class files. Embodiments remove duplicate information elements from a set of class files to reduce the size of individual class files and to prevent redundant resolution of the information elements. Memory allocation requirements are determined in advance for the set of classes as a whole to reduce the complexity of memory allocation when the set of classes are loaded. The class files are stored in a single package for efficient storage, transfer and processing as a unit. In an embodiment, a pre-processor examines each class file in a set of class files to locate duplicate information in the form of redundant constants contained in a constant pool. The duplicate constant is placed in a separate shared table, and all occurrences of the constant are removed from the respective constant pools of the individual class files. During pre-processing, memory allocation requirements are determined for each class file, and used to determine a total allocation requirement for the set of class files. The shared table, the memory allocation requirements and the reduced class files are packaged as a unit in a multi-class file.

45 Claims, 6 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 6

Full	Title	Citation	Front	Review	Classification	Date	Reference	Class. Index	Class. No.	Claims	KOMIC	Draw. No.
------	-------	----------	-------	--------	----------------	------	-----------	--------------	------------	--------	-------	-----------

☐ 6. Document ID: US 5966702 A

L9: Entry 6 of 6

File: USPT

Oct 12, 1999

US-PAT-NO: 5966702

DOCUMENT-IDENTIFIER: US 5966702 A

TITLE: Method and apparatus for pre-processing and packaging class files

DATE-ISSUED: October 12, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Fresko; Nedim	San Francisco	CA		
Tuck; Richard	San Francisco	CA		

US-CL-CURRENT: 707/1; 707/10, 707/7

ABSTRACT:

A method and apparatus for pre-processing and packaging class files. Embodiments remove duplicate information elements from a set of class files to reduce the size of individual class files and to prevent redundant resolution of the information elements. Memory allocation requirements are determined in advance for the set of

classes as a whole to reduce the complexity of memory allocation when the set of classes are loaded. The class files are stored in a single package for efficient storage, transfer and processing as a unit. In an embodiment, a pre-processor examines each class file in a set of class files to locate duplicate information in the form of redundant constants contained in a constant pool. The duplicate constant is placed in a separate shared table, and all occurrences of the constant are removed from the respective constant pools of the individual class files. During pre-processing, memory allocation requirements are determined for each class file, and used to determine a total allocation requirement for the set of class files. The shared table, the memory allocation requirements and the reduced class files are packaged as a unit in a multi-class file.

23 Claims, 6 Drawing figures
 Exemplary Claim Number: 1
 Number of Drawing Sheets: 6

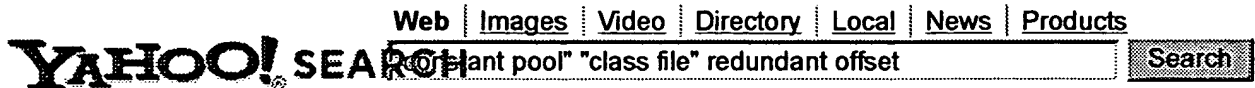
Full	Title	Citation	Front	Review	Classification	Date	Reference	Abstract	Publication	Claims	FIGS	Drawings
------	-------	----------	-------	--------	----------------	------	-----------	----------	-------------	--------	------	----------

Clear	Generate Collection	Print	Fwd Refs	Bkwd Refs	Generate OACS
-------	---------------------	-------	----------	-----------	---------------

Terms	Documents
L3 and (Jar file)	6

Display Format:

[Previous Page](#) [Next Page](#) [Go to Doc#](#)



[Shortcuts](#) [Advanced Search](#) [Preferences](#)

Search Results Results **1 - 10** of about **56** for **constant pool class file redundant offset** - 0.41 sec. (A

[:lass File Optimization](#)

ome. Introduction. Version History. Downloads ... a **class file** and you may be surprised by the space that the Cc / **redundant** FQNs, an obvious ... continue at the **offset** represented by 648 ...
www.demiclass.com/fulldoc.html - 25k - [Cached](#) - [More from this site](#)

[AZZ: An Efficient Compressed Format for Java Archive Files \(PDF\)](#)

of the. **class file**. The Java **constant pool** contains all the ... able table **offset**, a bytecode **offset** for branches, **ool** entries, ...
www.cs.purdue.edu/homes/jv/pubs/cascon.pdf - 73k - [View as html](#) - [More from this site](#)

<http://www.publicsource.apple.com/darwinsource/DevToolsJun2003/gcc-1435/gcc/java/java-tree>

Definitions for parsing and type checking for the GNU compiler for the Java(TM) language.
publicsource.apple.com/darwinsource/.../gcc-1435/gcc/java/java-tree.h - 70k - [Cached](#) - [More from this site](#)

<http://www.ittc.ku.edu/ANTS/challenge/ttrt/jdk1.2.2/include-old/jvm.h>

the byte code verifier and **class file** format checker ... cb); /* * Returns the **constant pool** types in the buffer ...
moving **redundant** separator characters ...
www.ittc.ku.edu/ANTS/challenge/ttrt/jdk1.2.2/include-old/jvm.h - 31k - [Cached](#) - [More from this site](#)

[iCC 3.4.0 - jcf-write - Best Web Site Sources](#)

out a Java(TM) **class file**. Copyright (C) 1998 ... value of the bytecode **offset** at the beginning of this ... word co
INDEX ...
www.best-websource.com/gcc/source-jcf-write.htm - 127k - [Cached](#) - [More from this site](#)

[index \(CMP Compiler Project Documentation\)](#)

containing a single **Constant Pool** entry for a class ... list of **constant pool** entries in a java **class file**. ... after r
redundant transition table states ...
www.chrisellsworth.com/cmp/index-all.html - 324k - [Cached](#) - [More from this site](#)

[each Yourself Java in 21 Days tyj21x.htm](#)

/ Charles L. Perkins. On today, your final day, the inner workings of the Java system will be revealed. ... for exam
le format ... there are and an **offset** that specifies how far into ... separately in all its (often **redundant**) detail. Lat
www.lipidat.chemistry.ohio-state.edu/java/tyj21x.htm - 143k - [Cached](#) - [More from this site](#)

[each Yourself Java in 21 Days tyj21x.html](#)

/ Charles L. Perkins. On today, your final day, the inner workings of the Java system will be revealed. ... for exam
le format ... there are and an **offset** that specifies how far into ... separately in all its (often **redundant**) detail. Lat
www.ashleyweb.net/JavaMan/tyj21x.html - 140k - [Cached](#) - [More from this site](#)

<http://open.homelinux.com/linuxdoc/gcc-java-3.2.2/ChangeLog>

Remove now-**redundant** code ... data_offset): Removed ... **redundant** set. * parse.y (java_expand_classes): N
compiled as local. * jcf-parse.c (parse_class_file ...
open.homelinux.com/linuxdoc/gcc-java-3.2.2/ChangeLog - 525k - [Cached](#) - [More from this site](#)

<http://cs.nyu.edu/~yap/unsup/unsup/installers/exact/gcc/gcc-2.95.3/gcc/java/ChangeLog>

arse.y (resolve_package): Fixed bogus return statement. (patch_method_invocation): Resolve method invocation
qualifier. *



Web Images Groups News Froogle Local^{New!} more »

"constant pool" "class file" redundant offset

Search

Advanced Search
Preferences

Web

Results 1 - 10 of about 130 for **"constant pool" "class file" redundant offset**. (0.34 seconds)

Brewing-up with the CAFE BABE

... can see no way of infecting a **.class file** without somehow ... that the routine used to parse the **constant pool** is very ... virus contains a great deal of **redundant code** ...

www.madchat.org/vxdevl/papers/avers/strange.html - 18k - [Cached](#) - [Similar pages](#)

[PDF] JAZZ: An Efficient Compressed Format for Java Archive Files ...

File Format: PDF/Adobe Acrobat - [View as HTML](#)

... the **constant pool**, the removal of **redundant constant pool** entries, and ... is the size of the **constant pool** index which ... on the semantics of the **class file** as long ...

www.cs.purdue.edu/homes/jv/pubs/cascon.pdf - [Similar pages](#)

[PDF] Announcements The JikesRVM Adaptive Optimization System JikesRVM ...

File Format: PDF/Adobe Acrobat - [View as HTML](#)

... SSA-based optimizations on scalars and arrays, **redundant load** and ... JikesRVM tables at the appropriate **offset** • Currently if ... In the **class file**, mark the ...

www.cs.ucsb.edu/~ckrintz/classes/w05/cs263/lectures/adaptonoff.pdf - [Similar pages](#)

[PDF] Binary Component Adaptation

File Format: PDF/Adobe Acrobat - [View as HTML](#)

... A table stores the byte **offset** of the missing **constant pool** index together with ... to a class, the delta file compiler parses the binary **class file**, looks up ...

www.cs.ucsb.edu/labs/oocsb/papers/TRCS97-20.pdf - [Similar pages](#)

[[More results from www.cs.ucsb.edu](#)]

[PDF] DataScript—a Specification and Scripting Language for Binary ...

File Format: PDF/Adobe Acrobat - [View as HTML](#)

... some file formats store length and **offset** parameters in ... that the "super class" field in a **class file** is either zero or the index of a **constant pool** en- try ...

www.stanford.edu/~gback/gback-datascript-saig2002.pdf - [Similar pages](#)

Java Virtual Machine's Internal Architecture

... designers may decide to store **redundant** information in ... The **constant pool** is described in more detail later in ... chapter and in Chapter 6, "The Java **Class File**." ...

www.artima.com/insidejvm/ed2/jvmp.html - 101k - [Cached](#) - [Similar pages](#)

The JVMPI Transition to JVMTI

... JVMPI implementations it was **redundant** with JVMPI_EVENT_DATA_DUMP_REQUEST other fields in the **class file** not shown here (max_stack , **constant pool**, etc ...

[java.sun.com/developer/technicalArticles/ Programming/jvmpitransition/](http://java.sun.com/developer/technicalArticles/Programming/jvmpitransition/) - 71k - [Cached](#) - [Similar pages](#)

[PDF] Complete Translation of Unsafe Native Code to Safe Bytecode

File Format: PDF/Adobe Acrobat - [View as HTML](#)

... The **.class file constant pool** size limit is still present, but it ... tion in the size of the ultimate **.class file**. ... both javac and jikes generate **redundant** bytecode ...

www.megacz.com/research/papers/nestedvm.ivme04.pdf - [Similar pages](#)

[PDF] Code Sharing among Virtual Machines

File Format: PDF/Adobe Acrobat - [View as HTML](#)

... otherwise, the runtime attempts to load the **class file**. ... way as shared symbols pointed to by the **constant pool**. ... the second indirection is **redundant**; thus, if it ...

www.aqute.biz/resources/embjava.ppt - Similar pages

Digitized by Google

Result Page: **1** **2** **3** **4** **5** **6** **7** **8** **9** **10** **Next**

Free! Get the Google Toolbar. [Download Now](#) - [About Toolbar](#)

Google Search Web PageRank 3 blocked AutoFill Options

"constant pool" "class file" redundan

[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied? Help us improve](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2005 Google



Web Images Groups News Froogle Local^{New!} more »

"constant pool" redundant offset portal

Search

Advanced Search
Preferences

Web

Results 1 - 4 of about 7 for "**constant pool**" **redundant offset portal**. (0.20 seconds)

Tip: Try removing quotes from your search to get more results.

[PDF] Fast, Effective Code Generation in a Just-In-Time Java Compiler

File Format: PDF/Adobe Acrobat

... Third, the code selector performs **redundant** load-after-store ... a single word (16 bits) for the **offset** and 16 ... and each object field has a unique **constant pool** index ...

portal.acm.org/ft_gateway.cfm?id=277740&type=pdf - [Similar pages](#)

[PDF] The Apprentice Challenge

File Format: PDF/Adobe Acrobat

... its immediate instance fields, a list of its static fields, its **constant pool**, a list ... created by the execution of the (NEW "Job") instruction at **offset** 11 below ...

portal.acm.org/ft_gateway.cfm?id=514189&type=pdf - [Similar pages](#)

United States Patent Application: 0040015852

... Similarly, the **redundant** FieldRef Bar entries 240-242 ... Accordingly, **constant pool** entries and other data/code within ... one another based on an **offset** value from ...

appft1.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&u=/nethtml/PTO/search-adv.html&r... - 75k

- Supplemental Result - [Cached](#) - [Similar pages](#)

United States Patent Application: 0020170047

... object code is identified by an **offset** contained in ... Foo entry 1203 in a shared **constant pool** 1202 within ... Similarly, the **redundant** FieldRef Bar entries 240-242 ...

appft1.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&u=%2Fnethtml%2FPTO%2Fsearch-adv.html... - 75k - Supplemental Result - [Cached](#) - [Similar pages](#)

[[More results from appft1.uspto.gov](#)]

In order to show you the most relevant results, we have omitted some entries very similar to the 4 already displayed.

If you like, you can repeat the search with the omitted results included.

Free! Google Desktop Search: Search your own computer. [Download now.](#)

Find: emails - files - chats - web history - media - PDF

"constant pool" redundant offset por

Search

[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied? Help us improve](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

Searching for **constant pool and offset and redundant**.

Restrict to: [Header](#) [Title](#) Order by: [Expected citations](#) [Hubs](#) [Usage](#) [Date](#) Try: [Google \(CiteSeer\)](#)
[Google \(Web\)](#) [Yahoo!](#) [MSN](#) [CSB](#) [DBLP](#)

Order: number of citations.

JAZZ: An Efficient Compressed Format for Java Archive Files - Bradley, Horspool, Vitek (1998) (Correct)
(13 citations)

component of a class file is typically the **constant pool**. It is not unusual for the **constant pool** to
Start-step-stop codes [1] are used for instruction **offsets** and string lengths they are also used to encode
used to encode the tables of Huffman codes. 5. **Redundant constant pool** entries are eliminated. Figure 1
www.csr.uvic.ca/~nigelh/Publications/jazz.pdf

Try your query at: [Google \(CiteSeer\)](#) [Google \(Web\)](#) [Yahoo!](#) [MSN](#) [CSB](#) [DBLP](#)

CiteSeer.IST - Copyright [Penn State](#) and [NEC](#)



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: ☒ The ACM Digital Library ☐ The Guide

+ "constant pool" + "class file" + offset + portal

SEARCH

THE ACM DIGITAL LIBRARY



[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used [constant pool](#) [class file](#) [offset](#) [portal](#)

Found 17 of 151,219

Sort results
by

relevance



[Save results to a Binder](#)

Try an [Advanced Search](#)

Display
results

expanded form



[Search Tips](#)

Try this search in [The ACM Guide](#)

☐ Open results in a new
window

Results 1 - 17 of 17

Relevance scale ☐ ☐ ☐ ☐ ☐

1 [JAZZ: an efficient compressed format for Java archive files](#)

Quetzalcoatl Bradley, R. Nigel Horspool, Jan Vitek

November 1998 **Proceedings of the 1998 conference of the Centre for Advanced Studies on Collaborative research**

Full text available: pdf (73.54 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The Jazz file format is intended to be a replacement for the JAR file format when used for storage and distribution of Java programs. A Jazz file is compressed to a degree that far exceeds what is possible with a JAR file. The smaller size of the Jazz format permits faster transmission speeds over a network and has the additional benefit of conserving disk storage. The compression is achieved as a combination of different data compression methods, adapted to suit the characteristics of collectio ...

2 [Compressing Java class files](#)

William Pugh

May 1999 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1999 conference on Programming language design and implementation**, Volume 34 Issue 5

Full text available: pdf (1.44 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Java class files are often distributed as jar files, which are collections of individually compressed class files (and possibly other files). Jar files are typically about 1/2 the size of the original class files due to compression. I have developed a wire-code format for collections of Java class files. This format is typically 1/2 to 1/5 of the size of the corresponding compressed jar file (1/4 to 1/10 the size of the original class files).

3 [Compact Java binaries for embedded systems](#)

Derek Rayside, Evan Mamas, Erik Hons

November 1999 **Proceedings of the 1999 conference of the Centre for Advanced Studies on Collaborative research**

Full text available: pdf (124.35 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Embedded systems bring special purpose computing power to consumer electronics devices such as smartcards, CD players and pagers. Java is being aggressively targeted at such systems with initiatives such as the Java 2 Platform, Micro Edition, which introduces certain efficiency optimizations to the Java Virtual Machine. Code size reduction has been identified as an important future goal for ensuring Java's success on embedded systems [20].


However, limited processing power and timing constraints ...

4 Practical extraction techniques for Java

Frank Tip, Peter F. Sweeney, Chris Laffra, Aldo Eisma, David Streeter

November 2002 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,

Volume 24 Issue 6

Full text available:  [pdf\(1.01 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Reducing application size is important for software that is distributed via the internet, in order to keep download times manageable, and in the domain of embedded systems, where applications are often stored in (Read-Only or Flash) memory. This paper explores extraction techniques such as the removal of unreachable methods and redundant fields, inlining of method calls, and transformation of the class hierarchy for reducing application size. We implemented a number of extraction techniques in < ...

Keywords: Application extraction, call graph construction, class hierarchy transformation, packaging, whole-program analysis

5 Multitasking without compromise: a virtual machine evolution

Grzegorz Czajkowski, Laurent Daynés

October 2001 **ACM SIGPLAN Notices , Proceedings of the 16th ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications**, Volume 36 Issue 11

Full text available:  [pdf\(220.97 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The multitasking virtual machine (called from now on simply MVM) is a modification of the Java virtual machine. It enables safe, secure, and scalable multitasking. Safety is achieved by strict isolation of application from one another. Resource control augment security by preventing some denial-of-service attacks. Improved scalability results from an aggressive application of the main design principle of MVM: share as much of the runtime as possible among applications and replicate everything el ...

Keywords: Java virtual machine, application isolation, native code execution, resource control

6 A scalable architecture for multi-threaded JAVA applications

M. Mrva, K. Buchenrieder, R. Kress

February 1998 **Proceedings of the conference on Design, automation and test in Europe**

Full text available:  [pdf\(117.76 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

 [Publisher Site](#)

The paper presents a scalable architecture for multi-threaded Java applications. Threads enable modeling of concurrent behavior in a more or less natural way. Thus threads give a migration path to multi-processor machines. The proposed architecture consists of multiple application-specific processing elements, each able to execute a single thread at one time. The architecture is evaluated by implementing a portable and scalable Java machine onto an FPGA board for demonstration.

Keywords: Java, application-specific, configurable, multi-threaded

7 Targeting GNAT to the Java virtual machine

Cyrille Comar, Gary Dismukes, Franco Gasperoni

8 Exploiting Java instruction/thread level parallelism with horizontal multithreading

Kenji Watanabe, Wanming Chu, Yamin Li

January 2001 **Australian Computer Science Communications , Proceedings of the 6th Australasian conference on Computer systems architecture**, Volume 23 Issue 4

Full text available:  [pdf\(787.34 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#)

Java bytecodes can be executed with the following three methods: a Java interpreter running on a particular machine interprets bytecodes; a Just-In-Time (JIT) compiler translates bytecodes to the native primitives of the particular machine and the machine executes the translated codes; and a Java processor executes bytecodes directly. The first two methods require no special hardware support for the execution of Java bytecodes and are widely used currently. The last method requires an embedded J ...

9 A java virtual machine architecture for very small devices

Nik Shaylor, Douglas N. Simon, William R. Bush

June 2003 **ACM SIGPLAN Notices , Proceedings of the 2003 ACM SIGPLAN conference on Language, compiler, and tool for embedded systems**, Volume 38 Issue 7

Full text available:  [pdf\(182.85 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The smallest complete Java™ virtual machine implementations in use today are based on the CLDC standard and are deployed in mobile phones and PDAs. These implementations require several tens of kilobytes. Smaller Java-like implementations also exist, but these involve compromises in Java semantics. This paper describes a JVM™ architecture designed for very small devices. It supports all the CLDC Java platform semantics, including exact garbage collection, dynamic class loading, and v ...

Keywords: CLDC, JVM, java, limited-memory devices, next generation smart cards

10 Recompilation for debugging support in a JIT-compiler

Mustafa M. Tikir, Jeffrey K. Hollingsworth, Guei-Yuan Lueh

November 2002 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 2002 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering**, Volume 28 Issue 1

Full text available:  [pdf\(89.55 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

A static Java compiler converts Java source code into a verifiably secure and compact architecture-neutral intermediate format, called Java *byte codes*. The Java byte codes can be either interpreted by a Java Virtual Machine or translated into native code by Java Just-In-Time compilers. Static Java compilers embed debug information in the Java class files to be used by the source level debuggers. However, the debug information is generated for architecture independent byte codes and most o ...

Keywords: Java, Java virtual machine debugger interface, debug information, dynamic recompilation, field access watch, just-in-time compilation

11 Automatic translation of Fortran to JVM bytecode

Keith Seymour, Jack Dongarra

June 2001 **Proceedings of the 2001 joint ACM-ISCOPE conference on Java Grande**

Full text available:

Additional Information:

This paper reports on the design of a FORTRAN-to-Java translator whose target language is the instruction set of the Java Virtual Machine. The goal of the translator is to generate Java implementations of legacy FORTRAN numerical codes in a consistent and reliable fashion. The benefits of directly generating bytecode are twofold. First, it provides a much more straightforward and efficient mechanism for translating FORTRAN GOTO statements. Second, it provides a framework for pursuing various ...

12 [Engineering a customizable intermediate representation](#)

K. Palacz, J. Baker, C. Flack, C. Grothoff, H. Yamauchi, J. Vitek

June 2003 **Proceedings of the 2003 workshop on Interpreters, Virtual Machines and Emulators**

Full text available:  pdf(322.87 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

The Ovm framework is a set of tools and components for building language runtimes. We present the intermediate representation and software design patterns used throughout the framework. One of the main themes in this work has been to support experimentation with new linguistic constructs and implementation techniques. To this end, framework components were designed to be parametric with respect to the instruction set on which they operate. We argue that our approach eases the task of writing new ...

13 [Middleware performance analysis: Performance monitoring of java applications](#)

M. Harkema, D. Quartel, B. M. M. Gijsen, R. D. van der Mei

July 2002 **Proceedings of the third international workshop on Software and performance**

Full text available:  pdf(219.69 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Over the past few years, Java has evolved into a mature platform for developing enterprise applications. A critical factor for the commercial success of these applications is end-to-end performance, e.g., in terms of response times, throughput and availability. This raises the need for the development, validation and analysis of performance models to predict performance metrics of interest. To develop and validate performance models, insight in the execution behavior of the application is essential ...

Keywords: performance measurement and monitoring of java applications

14 [Assembly language through the Java virtual machine](#)

Carl Bredlau, Dorothy Deremer

February 2001 **ACM SIGCSE Bulletin , Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education**, Volume 33 Issue 1


Full text available:  pdf(43.43 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Student understanding of choices and design decisions about instruction formats, addressing, instruction types, and flow of control is developed through the assembly level. We propose the Java Virtual Machine (JVM) as the architecture for teaching assembly language. The paper describes how to use the JVM as a teaching tool for a sophomore level Assembly Language and Computer Architecture course.

15 [Stacking them up: a comparison of virtual machines](#)

K John Gough

January 2001 **Australian Computer Science Communications , Proceedings of the 6th Australasian conference on Computer systems architecture**, Volume 23 Issue 4

Full text available:  pdf(694.50 KB) Additional Information:

A popular trend in current software technology is to gain program portability by compiling programs to an intermediate form based on an abstract machine definition. Such approaches date back at least to the 1970s, but have achieved new impetus based on the current popularity of the programming language Java. Implementations of language Java compile programs to bytecodes understood by the Java Virtual Machine (JVM). More recently Microsoft have released preliminary details of their ".NET" platform ...

16 [The apprentice challenge](#)

J. Strother Moore, George Porter

May 2002 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,

Volume 24 Issue 3

Full text available:  [pdf\(212.09 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We describe a mechanically checked proof of a property of a small system of Java programs involving an unbounded number of threads and synchronization, via monitors. We adopt the output of the javac compiler as the semantics and verify the system at the bytecode level under an operational semantics for the JVM. We assume a sequentially consistent memory model and atomicity at the bytecode level. Our operational semantics is expressed in ACL2, a Lisp-based logic of recursive functions. Our proofs ...

Keywords: Java, Java Virtual Machine, mutual exclusion, operational semantics, parallel and distributed computation, theorem proving

17 [Fast, effective code generation in a just-in-time Java compiler](#)

Ali-Reza Adl-Tabatabai, Michał Cierniak, Guei-Yuan Lueh, Vishesh M. Parikh, James M. Stichnoth

May 1998 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1998 conference on Programming language design and implementation**, Volume 33 Issue 5

Full text available:  [pdf\(1.44 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A "Just-In-Time" (JIT) Java compiler produces native code from Java byte code instructions during program execution. As such, compilation speed is more important in a Java JIT compiler than in a traditional compiler, requiring optimization algorithms to be lightweight and effective. We present the structure of a Java JIT compiler for the Intel Architecture, describe the lightweight implementation of JIT compiler optimizations (e.g., common subexpression elimination, register allocation, and elim ...

Results 1 - 17 of 17

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)